CLAIMS

1. (Canceled)

2. (Currently amended) A method comprising:

analyzing input terms on a bit-wise basis to segment each level of bit significance of input terms into one or more groups of three bits, and/or one or more groups of two bits and/or one or more groups of one bit;

designing a hyperpipelined series of Boolean function generators, based at least in part on the analyzing, to implement a Wallace-architecture of full-adders to receive at least a portion of the one or more groups of three bits, half-adders to receive at least a portion of the one or more of the groups of two bits, associated registers to receive at least a portion of the one or more the groups of one bit in the selected resources, and a multi-input adder to combine the input terms to produce intermediate summation results;

selecting atomic elements of a dedicated logic device in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers; ~~and~~

selecting atomic elements to implement control logic, the control logic configured to dynamically reallocate the atomic elements implementing the Wallace-architecture of full-adders, half-adders, and associated registers in response to a subsequent analysis of the input terms on a bit-wise basis;

selecting atomic elements to implement a first combinatorial module, the first combinatorial module to generate partial products of real components of a plurality of complex numbers; and

selecting atomic elements to implement a second combinatorial module, the second combinatorial module to generate partial products of imaginary components of the complex numbers;

wherein the partial products of the combinatorial modules form the input terms.

3.-4. (Canceled)

5. (Previously Presented) The method of claim 2 wherein selecting atomic elements of the dedicated logic device in which to implement the Wallace-architecture comprises selecting

atomic elements of a field programmable gate array (FPGA) in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

6. (Canceled)

7. (Currently amended) The method of claim 2, further comprising:

determining a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented; and

selecting atomic elements of the dedicated logic device ~~so as~~ to reduce a number of atomic elements left unused in the design, to implement the Wallace architecture with a number of full-adders, half-adders, and associated registers that approaches the minimum number.

8. (Currently amended) The method of claim 2 further comprising:

selecting atomic elements of the dedicated logic device to implement the Wallace-architecture of full-adders, half-adders, and associated registers; and

wherein designing the hyperpipelined series of Boolean function generators comprises assigning proximate atomic elements functions ~~so as~~ to reduce routing distances between stages of the Wallace-architecture.

9. (Previously Presented) The method of claim 2 further comprising:

determining a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented; and

wherein designing the hyperpipelined series of Boolean function generators comprises selecting atomic elements of the dedicated logic device to implement the Wallace-architecture with the minimal number of full-adders, half-adders, and associated registers while concurrently assigning proximate atomic elements to functions that result in reducing routing distances between stages of the Wallace-architecture.

10. (Previously Presented) The method of claim 2 further comprising:

selecting one or more field programmable gate arrays (FPGAs) in the system in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers;

wherein designing the hyperpipelined series of Boolean function generators to implement the Wallace-architecture of full-adders, half-adders, and associated registers in the selected resources comprises designing the hyperpipelined series of Boolean function generators to be implemented with the atomic elements of the selected FPGA(s).

11. (Previously presented) The method of claim 2 wherein designing the hyperpipelined series of Boolean function generators to implement the Wallace-architecture comprises designing the hyperpipelined series of Boolean function generators to increase grouping of bits of a same level of bit-significance of the input terms.

12. (Previously Presented) The method of claim 2 wherein designing the hyperpipelined series of Boolean function generators comprises dynamically designing the hyperpipelined series of Boolean function generators to implement desired instances of the Wallace-architecture in response to the control logic.

13. (Previously presented) The method of claim 2 further comprising:
implementing the design in the dedicated logic device by assigning the atomic elements of the dedicated logic device according to the design.

14. (Previously Presented) The method of claim 2 further comprising:

identifying features of other components that can be integrated into the design of the Wallace-architecture; and

wherein designing the hyperpipelined series of Boolean function generators includes integrating the identified features into the Wallace-architecture.

15. (Previously presented) The method of claim 14 wherein designing the hyperpipelined series of Boolean function generators includes adding accumulator bits from the other components to a summation result achieved by the Wallace-architecture.

16. (Previously presented) The method of claim 14 wherein designing the hyperpipelined series of Boolean function generators includes coupling the Wallace-architecture to output a summation result to at least one of the other components.

17. (Currently amended) An article of manufacture comprising a machine-accessible medium having content to provide instructions for generating a complex arithmetic summing module, the content to provide the instructions to cause an electronic system to:

analyze input terms on a bit-wise basis to segment each level of bit significance of input terms into one or more groups of three bits, and/or one or more groups two bits and/or one or more groups of one bit;

design a hyperpipelined series of Boolean function generators to implement a Wallace-architecture of full-adders, based at least in part on the analyzed input terms, to receive at least a portion of the one or more groups of three bits, half-adders to receive at least a portion of the one or more the groups of two bits, associated registers to receive at least a portion of the one or more the groups of one bit in the selected resources, and a multi-input adder to combine the input terms to produce intermediate summation results;

select atomic elements of a dedicated logic device in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers; ~~and~~

select atomic elements to implement control logic, the control logic configured to dynamically reallocate the atomic elements implementing the Wallace-architecture of full-adders, half-adders, and associated registers in response to subsequent analysis of the input terms on a bit-wise basis;

select atomic elements to implement a first combinatorial module, the first combinatorial module to generate partial products of real components of a plurality of complex numbers; and

select atomic elements to implement a second combinatorial module, the second combinatorial module to generate partial products of imaginary components of the complex numbers;

wherein the partial products of the combinatorial modules form the input terms.

18.-19. (Canceled)

20. (Previously Presented) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to select atomic elements of a field programmable gate array (FPGA) in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers.

21. (Canceled)

22. (Currently amended) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to:

determine a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented; and

select atomic elements of the dedicated logic device ~~so as~~ to reduce a number of atomic elements left unused in the design, to implement the Wallace architecture with a number of full-adders, half-adders, and associated registers that approaches the minimum number.

23. (Currently amended) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to:

provide instructions to cause the electronic system to assign proximate atomic elements functions ~~so as~~ to reduce routing distances between stages of the Wallace-architecture.

24. (Previously Presented) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to:

determine a minimal number of full-adders, half-adders, and associated registers with which the Wallace-architecture could be implemented; and

select atomic elements of the dedicated logic device to implement the Wallace-architecture with the minimal number of full-adders, half-adders, and associated registers and concurrently to assign proximate atomic elements to functions to result in reducing routing distances between stages of the Wallace-architecture.

25. (Previously Presented) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to:

select one or more field programmable gate arrays (FPGAs) coupled with the controller in which to implement the Wallace-architecture of full-adders, half-adders, and associated registers; and

design the hyperpipelined series of Boolean function generators to be implemented with the atomic elements of the selected FPGA(s).

26. (Previously Presented) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to design the hyperpipelined series of bits of a same level of bit-significance of the input terms.

27. (Previously Presented) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to dynamically design the hyperpipelined series of Boolean function generators to implement desired instances of the Wallace-architecture.

28. (Previously Presented) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to:

implement the design in the dedicated logic device by assigning the atomic elements of the dedicated logic device according to the design.

29. (Previously Presented) The article of manufacture of claim 17 further comprising content to provide instructions to cause the electronic system to:

identify features of the other components that can be integrated into the design of the Wallace-architecture; and

wherein the content to provide instructions to cause the electronic system to design the hyperpipelined series of Boolean function generators includes the content to provide instructions to cause the electronic system to integrate the identified features into the Wallace-architecture.

30. (Previously Presented) The article of manufacture of claim 29 further comprising content to provide instructions to cause the electronic system to add accumulator bits from the other components to a summation result achieved by the Wallace-architecture.

31. (Previously Presented) The article of manufacture of claim 29 further comprising content to provide instructions to cause the electronic system to couple the Wallace-architecture to output a summation result to at least one of the other components.

32. (Previously presented) The method of claim 2, wherein designing the hyperpipelined series of Boolean function generators further comprises:

designing the hyperpipelined series of Boolean function generators to implement the multi-input adder in a pipeline stage of the hyperpipelined series of Boolean function generators immediately after a final pipeline stage of full-adders, half-adders, and/or registers.

33. (Canceled)

34. (Currently amended) The method of claim ~~34~~ 2, further comprising:

selecting atomic elements to implement a third combinatorial module, the third combinatorial module to generate partial products of a real component of a first complex number of complex numbers and an imaginary component of a second complex number of the complex numbers; and

selecting atomic elements to implement a fourth combinatorial module, the fourth combinatorial module to generate partial products of an imaginary component of the first complex number of complex numbers and a real component of the second complex number of the complex numbers.

35. (Currently amended) A logic device, comprising:

a plurality of hybrid summing modules to combine a plurality of partial products into a plurality of intermediate summation results;

~~an~~ plurality of adders to add ~~the~~ corresponding ones of the plurality of intermediate summation results; ~~and~~

control elements to dynamically reallocate a plurality of atomic elements forming at least one of the hybrid summing modules and the adder in response to an analysis of the partial products;

a first combinatorial module to generate a first portion of the partial products in response to a first input and a second input; and

a second combinatorial module to generate a second portion of the partial products in response to a third input and a fourth input.

36. (Canceled)

37. (Currently amended) The logic device of claim ~~36~~ 35, the hybrid summing module being a first hybrid summing module and the adder being a first adder, the logic device further comprising:

a third combinatorial module to generate a first portion of a second plurality of partial products in response to the first input and the fourth input;

a fourth combinatorial module to generate a second portion of the second plurality of partial products in response to the second input and the third input;

a second hybrid summing module to combine a plurality of partial products into a plurality of intermediate summation results; and

a~~n~~ second adder to add the intermediate summation results.